



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/017,183	12/07/2001	Coach Wei	EMC-06-235(PRO)ORD1	5883
24227 7590 05/28/2008 EMC CORPORATION OFFICE OF THE GENERAL COUNSEL 176 SOUTH STREET HOPKINTON, MA 01748			EXAMINER ZHEN, L I B	
			ART UNIT 2194	PAPER NUMBER
			MAIL DATE 05/28/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/017,183

Applicant(s)

WEI, COACH

Examiner

Li B. Zhen

Art Unit

2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 February 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 and 22-25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-18 and 22-25 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/S508)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. Claims 1 – 18 and 22 – 25 are pending in the current application.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 2/19/2008 has been entered.

Response to Arguments

3. Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. **Claims 1-18 and 22-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,901,554 to Bahrs et al. [hereinafter Bahrs,**

previously cited] in view of “What are Enterprise JavaBeans components?: Part 1: The history and goals of EJB architecture” [hereinafter Nordby, previously cited] and further in view of U.S. 6,064,382 to Diedrich et al. [hereinafter Diedrich].

6. As to claim 1, Bahrs teaches the invention substantially as claimed including a method for delivering applications over a network [business logic; col. 31, lines 5 – 15 and col. 14, lines 23 – 36] from a the backend server [a server 104; col. 12, lines 16 – 43; server side business logic, col. 31, lines 5 - 15] to a plurality of client devices [clients 108, 110, and 112; col. 12, lines 16 – 43], the method comprising the steps of:

receiving a request from a client and determining a type of the client [placing and presenting views in a client, for issuing requests for different concurrent servers and services from the client; col. 14, lines 36 – 65];

having the application invoke a Graphical User Interface (GUI) Application Programming Interface (API) to present the application's user interface [a client browser invokes a URL submit, the web server obtains the request and passed control to a servlet. The servlet obtains a key/value pair list of values entered in the HTML client. This list is passed to the ViewController alternate view being displayed; col. 38, lines 1 – 19];

replacing the GUI API with a re-implemented [replacement may be accomplished by creating the developer's own implementation of ViewControllerBaseImpl that implements the methods getComponent(), setEnabled(boolean enable), and setVisible(boolean visible), col. 20, lines 33 – 52; overriding methods of the

Art Unit: 2194

ViewController class, col. 31, line 53 – col. 32, line 22; examiner notes that when the methods of the ViewController class is overridden with the developer own implementation, the View Controller class is re-implemented] network aware GUI API [ViewController interface 3902 extends JTC interface 3904; col. 35, lines 45 – 54 and col. 44, line 13 – 50] that describe the Graphical User Interface [col. 48, lines 40 – 60 and col. 53, lines 3 – 20], event processing registries [data is passed via different events, such as ViewEvent 510, RequestEvent 522, and RequestEvent 526; col. 17, lines 25 – 39] and other related information [object handling placement of components will register as a listener for notifications to place objects on the screen; col. 24, lines 36 – 59] corresponding to a presentation layer of the application in a high level, object level messages [col. 16, line 57 – col. 17, line 15];

sending such messages to the client device via the network [col. 41, line 66 – col. 42, line 19; col. 48, lines 40 – 60 and col. 53, lines 3 – 20];

processing the messages and rendering a user interface by a client-side program operating at the client [If the major code for the TopEvent is message, then the message is displayed for the application (step 8418); col. 49, lines 25 – 33], which delivers a user experience for that device according to the capability of the client [mechanism for creating the HTML view is application dependent/screen dependent; col. 37, line 50 – 67];

rendering the user interface on the client device [ViewController 502 basically provides a reusable GUI element; col. 15, line 52 – col. 16, line 13];

transmitting a plurality of user input and client-side events back to the server [col. 36, lines 17 – 28] via a predetermined protocol [col. 14, lines 36 – 65];

processing the user input and client-side events on the backend server [col. 26, lines 1 – 20 and col. 16, line 56 – col. 17, line 15], translating such events and inputs as if they were locally generated [ViewEvents generated in the ViewControllers 12302 being handled by the ApplicationMediator 12304 and translated into appropriate RequestEvents; col. 65, lines 23 – 41], and sending such translated events and inputs to the application for processing [RequestEvents are passed on to the destination 12308 via the transported 12306; col. 65, lines 23 – 41];

encoding and routing the output of the application to the client device using the predetermined messaging format [col. 16, line 57 – col. 17, line 15]; and

further processing the output by the client-side program to refresh the Graphical User Interface [the return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user; col. 16, line 57 – col. 17, line 15]. Bahrs discloses that the ViewControllerImpl that implements the ViewController and JTC interfaces is usually a Java Component or Container or bean [col. 19, lines 42 – 56]. Bahrs does not specifically disclose applications that are developed once and deployed multiple times.

However, Nordby teaches an EJB component can be developed once and then deployed on multiple platforms without recompilation or source code modification [p. 4, EJB Technology design goals].

Bahrs teaches that the ViewControllerImpl that implements the ViewController and JTC interfaces is usually a Java Component or Container or bean. It would have

been obvious to a person of ordinary skill in the art at the time the invention was made to implement the ViewControllerImpl of Bahrs as a Java bean and provide applications that can be developed once and deployed multiple times because this simplifies development of middleware components that are transactional, scalable, and portable [p. 1, 4th paragraph of Nordby] and provides a robust, scalable environment that can support mission-critical enterprise information systems [p. 1, 5th paragraph of Nordby]. Bahrs as modified does not teach a plurality of client devices, at least two of the client devices differing in type and display capabilities, receiving a request from a client and determining a type of the client, in response to the type of the client replacing the GUI API with a re-implemented network aware GUI API comprising a User Interface (UI) record, the record comprising pre-determined format based messages that that describe the Graphical User Interface, processing the messages in the UI record and rendering a user interface by a client-side program operating at the client, which delivers a user experience for that device according to the display capability of the client.

However, Diedrich teaches a plurality of client devices [col. 2, line 60 – col. 3, line 10], at least two of the client devices differing in type and display capabilities [display file 250 typically includes the definition of one or more formats; col. 8, lines 50 – 60], receiving a request from a client and determining a type of the client [API then determines which display file 250 is specified; col. 5, lines 46 – 60], in response to the type of the client replacing the GUI API with a re-implemented network aware GUI API comprising a User Interface (UI) record [Green screen application 126 directly calls GUI APIs 128 by virtue of replacing all its display I/O function calls 243 with calls 443 to GUI

APIs 128; col. 8, lines 6 – 60], the record comprising pre-determined format based messages that describe the Graphical User Interface [display file may be specified along with the variable display data, but more commonly is specified by a separate FORMAT command that identifies the appropriate display file 250 before passing the variable display data 242 to the PUT or PUTGET API; col. 5, lines 45 – 60], processing the messages in the UI record and rendering a user interface by a client-side program operating at the client, which delivers a user experience for that device according to the display capability of the client [Once the classes have been modified as required, the screen classes are compiled and made available to the client; col. 8, lines 60 – col. 9, lines 10].

It would have been obvious to a person of ordinary skill in the art to further modify the invention of Bahrs and Nordby to incorporate the features of Diedrich. One of ordinary skill in the art would have been motivated to make the combination because this provides a graphical user interface for existing host-based applications by defining some object oriented classes that reside on the client workstation, and by substituting function calls for display data in the screen application with function calls that interface with the object oriented GUI defined by the classes [col. 2, line 60 – col. 3, line 10].

7. As to claim 22, Bahrs as modified teaches a system for distributing an application [col. 14, lines 23 – 36 of Bahrs] to a plurality of client devices having different display capabilities [col. 8, lines 50 – 60 of Diedrich] includes at least a server [a server 104; col. 12, lines 15 – 45 of Bahrs], at least a client device [clients 108, 110, and 112; col.

12, lines 16 – 43 of Bahrs], and a communication means [network 102; col. 12, lines 16 – 45 of Bahrs], the system comprising:

a presentation layer of the application [ViewController; col. 15, line 52 – col. 16, line 12 of Bahrs] written using a server-side API [col. 19, lines 12 – 30 of Bahrs] based network programming model [col. 28, lines 42 – 67 of Bahrs];

a business logic layer of the application [business logic; col. 31, lines 5 – 15 and col. 14, lines 23 – 36 of Bahrs] and a data layer of the application [data model; col. 35, line 57 – col. 36, line 6 of Bahrs] both of which are written with the server-side API and running on the server [a server 104; col. 12, lines 16 – 43; server side business logic, col. 31, lines 5 - 15 of Bahrs]; and where

the server-side API having a supporting infrastructure that: sends [Object data may take various forms, such as Extensible Markup Language (XML), String, Hypertext Markup Language (HTML), key/value, Remote Method Invocation (RMI), J/XFS, RS232; col. 17, lines 25 – 39 of Bahrs] different User Interface (UI) records comprising information [col. 8, lines 6 – 60 of Diedrich] associated with the application's user interface information [col. 47, line 63 – col. 48, line 15 of Bahrs] to a plurality of client devices [col. 48, lines 40 – 60 and col. 53, lines 3 – 20 of Bahrs and col. 2, line 60 – col. 3, line 10 of Diedrich], each UI record modifying the application's user interface according to the display capabilities of the respective client to enable display of a modified version of the application's user interface by the respective client [col. 5, lines 45 – 60 and col. 8, lines 6 – 60 of Diedrich], handles communications problems [col. 43, lines 15 – 36 of Bahrs], renders the application's user interface [ViewController 502

Art Unit: 2194

basically provides a reusable GUI element; col. 15, line 52 – col. 16, line 13 of Bahrs], dispatches necessary user input events back to the server for processing [col. 18, line 63 – col. 19, line 13 of Bahrs]; and

wherein use of the system enable the application and the application user interface [col. 19, lines 42 – 56 of Bahrs and col. 8, lines 60 – col. 9, lines 10 of Diedrich] to be developed once and deployed multiple times by different types of client devices [EJB component can be developed once and then deployed on multiple platforms without recompilation or source code modification; p. 4, EJB Technology design goals of Nordby].

8. As to claim 23, Bahrs as modified teaches an apparatus for distributing an application over a network [col. 14, lines 23 – 36 of Bahrs] to a plurality of client devices having different display capabilities [col. 8, lines 50 – 60 of Diedrich] where the apparatus includes:

a server [a server 104; col. 12, lines 15 – 45 of Bahrs];

a network communication means [network 102; col. 12, lines 16 – 45 of Bahrs];

a storage device for storing, for each client device of the plurality of client devices, a User Interface (UI) record [col. 8, lines 6 – 60 of Diedrich] associated with a re-implemented [replacement may be accomplished by creating the developer's own implementation of ViewControllerBaseImpl that implements the methods getComponent(), setEnabled(boolean enable), and setVisible(boolean visible), col. 20, lines 33 – 52 of Bahrs; overriding methods of the ViewController class, col. 31, line 53 –

col. 32, line 22 of Bahrs; examiner notes that when the methods of the ViewController class is overridden with the developer own implementation, the View Controller class is re-implemented] network based API module that is used to transparently replace the API on which the application was developed [ViewController interface 3902 extends JTC interface 3904; col. 35, lines 45 – 54 and col. 44, line 13 – 50 of Bahrs] and is customized according to display capabilities of the respective client device [col. 5, lines 45 – 60 and col. 8, lines 6 – 60 of Diedrich];

a first means for running an application of the plurality of applications where a business logic [business logic; col. 31, lines 5 – 15 and col. 14, lines 23 – 36 of Bahrs] of the application runs on the server [a server 104; col. 12, lines 16 – 43; server side business logic, col. 31, lines 5 - 15 of Bahrs];

a second means for forwarding a given UI record to a client in response to a launch of the application by the client to display the application interface on the client device in accordance with display capabilities of the client device [col. 2, line 60 – col. 3, line 10 of Diedrich];

a third means for transferring the user interactions on the client device to the server [col. 18, line 63 – col. 19, line 13 of Bahrs], calculating the appropriate response to the input [deliver the information to the server's service for processing; col. 16, line 56 – col. 17, line 15 of Bahrs], and transmitting the appropriate response to the client machine [response data will be returned to the Transporter 524 in a RequestEvent; col. 16, line 56 – col. 17, line 15 of Bahrs];

a fourth means for updating the display of the application on the client device based on the responses from the server [return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user; col. 16, line 56 – col. 17, line 15 of Bahrs];

wherein use of the re-implemented network aware API enables the application [col. 19, lines 42 – 56 of Bahrs and col. 8, lines 60 – col. 9, lines 10 of Diedrich] and application interface to be developed once and deployed multiple times on different client devices having different display capabilities [EJB component can be developed once and then deployed on multiple platforms without recompilation or source code modification; p. 4, EJB Technology design goals of Nordby].

9. As to claim 2, Bahrs teaches the GUI API and the event processing API are represented as classes within Java Foundation Classes [col. 14, lines 36 – 65].

10. As to claim 3, Bahrs teaches the client-side program is a computer program based on Operating System's API [col. 34, lines 30 – 39 and col. 13, lines 43 – 60].

11. As to claim 4, Bahrs teaches the client-side program is a wireless device program written using the device's Operating System's API [col. 15, lines 26 – 52 and col. 14, lines 1 – 17].

Art Unit: 2194

12. As to claim 5, Bahrs teaches the client-side program is program written using Java API [col. 14, lines 36 – 65 and col. 15, lines 25 – 52].

13. As to claim 6, Bahrs teaches the JAVA API is selected from the group consisting of: Abstract Windows Toolkit (AWT), Personal Java, Java 2 Micro Edition based GUI API or Java Swing [col. 14, lines 36 – 65 and col. 35, lines 45 – 54 and col. 44, line 13 – 50].

14. As to claim 7, Bahrs teaches the predetermined protocol is Hyper Text Transfer Protocol (HTTP) [JTC has natural support for multiple protocols, such as, for example IIOP, RMI, Sockets, HTTP, HTTPs, and Files; col. 15, lines 26 – 52].

15. As to claim 8, Bahrs teaches the predetermined protocol is Hyper Text Transfer Protocol over Secure Socket Layer (HTTPS) [JTC has natural support for multiple protocols, such as, for example IIOP, RMI, Sockets, HTTP, HTTPs, and Files; col. 15, lines 26 – 52].

16. As to claim 9, Bahrs as modified teaches wireless devices [col. 15, lines 26 – 52 of Bahrs] and the Wireless Application Protocol (WAP) protocol [col. 8, lines 15 – 27 of Diedrich].

17. As to claim 10, Bahrs as modified teaches the predetermined protocol is proprietary [col. 5, lines 5 – 23 of Diedrich].

18. As to claim 11, Bahrs teaches the predetermined messaging format is based on Extended Markup Language (XML) [col. 17, lines 25 – 38 and col. 37, line 50 – 67].

19. As to claim 12, Bahrs as modified teaches the predetermined messaging format is proprietary [col. 5, lines 5 – 23 of Diedrich].

20. As to claim 13, Bahrs teaches the network is the Internet [col. 12, lines 16 – 43].

21. As to claim 14, Bahrs teaches the network is a local area network [col. 12, lines 16 – 43].

22. As to claim 15, Bahrs teaches the local area network is a bandwidth-limited slow speed network [col. 1, line 58 – col. 2, line 15].

23. As to claim 16, Bahrs teaches the network includes a wireless network [col. 15, lines 25 – 52].

Art Unit: 2194

24. As to claim 17, Bahrs teaches the client device is selected from the group consisting of workstations, desktops, laptops, Person Digital Assistants (PDAs), and wireless devices [col. 15, lines 25 – 52].

25. As to claim 18, Bahrs teaches the server and the client device are combined into one entity [col. 17, lines 61 – 67 and col. 31, lines 5 – 15].

26. As to claim 24, Bahrs teaches the application code is not modified when distributing the application [col. 14, lines 23 – 36] and the application code is not distributed to the client device [business logic and central data management of an application should be separated out from the JTC application; col. 31, lines 5 – 15].

27. As to claim 25, Bahrs teaches distributing a plurality of pre-existing applications [col. 14, lines 23 – 36].

CONTACT INFORMATION

28. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2194

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Li B. Zhen
Primary Examiner
Art Unit 2194

/Li B. Zhen/
Primary Examiner, Art Unit 2194